Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L18: Entry 2 of 2 File: USPT Dec 2, 1997

DOCUMENT-IDENTIFIER: US 5694603 A
TITLE: Computer memory product with preemptive multithreading software

Application Filing Date (1):
19900320

Brief Summary Text (2):
This invention relates to computer systems, and more particularly, to a novel computer memory product including program software interrupt-driven preemptive multithreaded processing so as to enable, for example, providing real-time processing of language and other alphanumeric code concurrently as the code is being entered or edited at the console.

Brief Summary Text (5):
A software program comprises a plurality of executable instruction sequences each having direct access to the same program address space. The execution of each sequence is termed a "thread". A clock or timer periodically activates an interrupt operation of the central processor. Each interrupt preempts an executing thread after the thread has executed for a brief timeslice during which the thread may have performed only a portion of its task. Control of the processor is thereby taken away from the preempted thread, and control then passes to an interrupt service routine which then passes control to another thread to invoke the latter for execution during the next timeslice. Control of the processor is thereafter returned to the preempted thread to enable the latter to resume execution at the point where it was previously interrupted. The threads are thus executed concurrently with each thread executed in its respective series of successive spaced timeslices. Control of the central processor is thus transferred repeatedly back and forth between the threads so rapidly that the threads are run substantially simultaneously and are so perceived by the user. The threads may thus execute incrementally and piecewise with their successive task portions executed alternately in a mutually interleaved relation and with each thread executed during its respective series of spaced timeslices interleaved with the timeslices of at least one other thread.

Brief Summary Text (8):
BACKGROUND OF THE ILLUSTRATIVE EMBODIMENT

Brief Summary Text (16):
In recent years the large increase in software costs, the lack of skilled programmers, the rapid expansion of the computer market, the widespread adoption of microcomputers, and the underutilization of much available hardware because of lack of software, have compelled the adoption of high-level languages and concerted efforts to make their use more efficient.

Brief Summary Text (21):
The programmer invokes the real-time compiler-editor system by typing its command file name at the keyboard console. The command file containing the software portion of the system is then read into memory from a disk. The source buffer, a memory region which is to contain the source code, is initialized so that its first stored

byte is a predetermined code which will be called a "Pause Mark". Execution of the compiler then begins by reading the Pause Mark as the character in the first location of the buffer. When the compiler reads the Pause Mark it will enter an infinite loop repeatedly reading the same location until the content of this location is changed by the editor to a blank (space).

Brief Summary Text (33):
It will be obvious to those skilled in the computer arts that the same or equivalent hardware and software may be employed to perform lexical and syntactic analyses of natural language code, or to calculate the numeric code of a spreadsheet, concurrently in real time as the user enters or edits such code at the keyboard.

Detailed Description Text (2):
The following is a detailed description of a preferred embodiment of the invention. The disclosed details are merely illustrative of one of the many forms which the invention may take in practise. The invention and novelty reside in neither the hardware nor the software taken separately, but rather in the novel combination of both.

Detailed Description Text (9):
The interrupt facility enables the programmer to stop execution of the machine code program at any time, examine the values of the variables, and then continue execution. No additional hardware is required for this extra function, and the extra software is minimal.

Other Reference Publication (4):
Collin, A. J. T., The Implementation of STAB-1, Software--Practice and Experience, vol. 2, 1972, pp. 137-142.

Other Reference Publication (5):
Artym, Richard, The STAB Multiprocessing Environment for CYBA-M, Software--Practice and Experience, vol. 12, 1982, pp. 323-329.

Other Reference Publication (7):
Duffie, C. A. III, Task Scheduling Algorithm for a Teleprocessing Communications Controller, IBM Technical Disclosure Bulletin, vol. 16, No. 10, Marcy 1974, pp. 3349-3352.

CLAIMS:

1. A computer-readable disk memory having a surface formed with a plurality of binary patterns constituting a multithreaded application program executable by a desktop computer having a central microprocessor, a memory, means for loading said application program into a defined address space of said memory, and a clock-driven periodically-activated interrupt operation providing a plurality of series of spaced timeslices with the timeslices of each series interleaved with the timeslices of at least one other series, said multithreaded program comprising

a plurality of sets of instructions with each set executable by said microprocessor to provide a respective thread of execution and with each thread having a respective task to perform,

a first of said sets of instructions executable to provide a first thread of execution having control of the central microprocessor during each successive timeslice of a first series of timeslices with successive portions of the task of said first thread performed during respective successive timeslices of said first series,

said first thread of execution being periodically preempted in response to

activations of said interrupt operation at predetermined fixed time intervals at a fixed frequency in the range of about every ten to thirty milliseconds so as to provide a preemption at each termination of a timeslice of said first series by said clock activation of said interrupt operation, and

a second of said sets of instructions executable to provide a second thread of execution and responsive to said periodic preemptions to acquire control of the central microprocessor during successive timeslices of a second series of timeslices with successive portions of the task of said second thread performed during respective successive timeslices of said second series,

whereby a preemptive multithreading mode of operation is provided for the concurrent execution of a plurality of instruction threads of the same program with each thread executing successive incremental portions of its task during successive timeslices of a respective series of spaced timeslices and with the successive executed task portions of each thread interleaved with the successive executed task portions of at least one other thread so as to provide concurrent execution of a plurality of threads of the same program,

each of said threads having direct access to said program memory address space so as to provide fast efficient preemption of one thread by another thread and switching of control of the central microprocessor back and forth among the threads at a rate so rapid that the threads execute effectively simultaneously,

thereby enabling a single microprocessor to simulate the parallel processing of a large complex mainframe computer having multiple central processing units.

2. A disk memory as set forth in claim 1 to provide interactive multithreading with a desktop computer having a user operable input device and a video display monitor, and wherein

one of said threads comprises instructions executable immediately in response to said user operable input device for interactively controlling the execution of said one thread,

said one thread further comprising instructions for activating said display monitor to display the effects of said interactive control by said user operable input device,

whereby said one thread executes interactively with the user in the foreground while another thread of the same program executes in the background concurrently with said one thread with control of the central microprocessor repeatedly switching back and forth between the threads, and

wherein said fast efficient preemption provided by said direct access by the threads to said memory address space enables control of the central microprocessor to be switched between the threads so rapidly that an interactive user perceives the foreground and background threads to be executing simultaneously and without any perceptible interference by the background thread with the user's interaction with the foreground thread.

3. A disk memory as set forth in claim 1 to provide interactive multithreading with a desktop computer having an interactive user-operable input device for transmitting information from an interactive user and a monitor screen for displaying information to the user, wherein

said second set of instructions comprises instructions executable immediately in response to said user-operable input device for entering said transmitted information and for interactively displaying on said screen the effects of said entered information,

Durations of pulses on LSOF and BOS are 50 nanoseconds or one (IDLC, L1) machine clock cycle. Each frame slot is allocatable to a communication channel, and each communication channel may be dynamically allocated one or more slots per frame. Active B, D, and clear/voice channels, are each allotted one slot per frame, and several slots (not necessarily contiguous in time) may be allotted collectively to a single HyperChannel (see description below of HyperChannels).

Detailed Description Text (537):
When an RDCR2 space becomes filled with 4 bytes, RFM posts a request to the DMARQ (DMA Request Queue) which causes the DMAC partition operating (asynchronously) in association with MIO partition to transfer the 4 bytes to a space in external (IOP/host) memory designated by address information in the respective RDCR1 field named RDCA (Receive DMA Current Address; 22 bits). The external memory is addressed as a circular buffer (refer to DMAC detailed description which follows). The remaining 10 bit spaces in RDCR1 consist of 8 reserved bit spaces (RES), a 1-bit space RPE (Receive Parity Error) for indicating parity error detected at the FIFOR output to DMAC, and a 1-bit space RBC (Receive Boundary Check) for indicating boundary check conditions encountered during the transfer (refer to DMAC description).

Detailed Description Text (1119):
If a valid address for a partition was decoded, enable select line (SIO.sub.-- DMACR.sub.-- RD, SIO.sub.-- DMACR.sub.-- WR, SIO.sub.-- FIFO.sub.-- RD, SIO.sub.-- FIFO.sub.-- WR, SIO.sub.-- TSR.sub.-- RD, SIO.sub.-- TSR.sub.-- WR, SIO.sub.-- RSM.sub.-- RD, or SIO.sub.-- RSM.sub.-- WR) to appropriate partition, based on the address decoded and the state of IOP.sub.-- RD/WR signal; go to state 3.

Detailed Description Text (1390):
The device also allows for synchronous storage of event status information associated with said requests in a memory shared for storing control parameters governing the processing activities in said pipeline. The device also allows for queued maintenance of event status in said memory, relative to plural time spaced events occurring within a communication channel in said shared memory, under the management direction of said interrupt handling partition. Finally, the device includes a slave I/O partition allowing the IOP to asynchronously access and retrieve said queued status with minimal interference to ongoing data communication processes.

Previous Doc Next Doc Go to Doc#

said first set of instructions comprises instructions executable to process said entered information concurrently in real time as the information is being transmitted by said user-operable input device, and

whereby said second thread executes in the foreground interactively with and visibly to the user while concurrently therewith said first thread executes in the background,

said control of the central microprocessor switching among the threads so rapidly that the user perceives that both threads appear to be executing simultaneously and without any perceptible interference by the background thread with the user's interaction with the foreground thread.

9. A memory as set forth in claim 8 for use with a desktop computer having an interactive user operable device, and wherein

one of said threads comprises instructions executable interactively with a user in response to said user operable device for interactively controlling the execution of said one thread,

whereby said one thread executes interactively with the user in the foreground while another thread of the same program executes in the background concurrently with said one thread with control of the central microprocessor repeatedly switching back and forth between the threads so rapidly that an interactive user perceives the foreground and backgrounds to be executing simultaneously.

10. A memory as set forth in claim 8 for use with a desktop computer having a user-operable device for transmitting information from an interactive user, and wherein

said second set of instructions comprises means interactive with a user and responsive to said user-operable device for entering said transmitted information, and

said first set of instructions comprises means to process said entered information concurrently in real time as the information is being transmitted by said user-operable device,

whereby said second thread executes in the foreground while concurrently therewith said first thread executes in the background so that the user perceives that both threads appear to be executing simultaneously while control of the central microprocessor is rapidly switching back and forth among the plurality of threads of the same program.

15. A disk means as set forth in claim 12 for use with a computer having a display and wherein said program comprises

operator responsive means for interactively controlling the execution of at least one of said program threads during the execution of said one thread, and

means for controlling said display for interactively showing to the operator the effects of said operator responsive control,

whereby a thread of a program may execute interactively with the operator in the foreground while another thread of the same program executes in the background concurrently with the interactive execution of the foreground thread so that an interactive operator perceives the foreground and background threads to be executing simultaneously.

16. A computer software memory element encoded with a plurality of binary patterns

constituting a multithreaded program having a plurality of threads of instructions executable by a computer having a single central processor, a clock-driven periodically-activated interrupt operation providing a plurality of series of spaced timeslices with the timeslices of each series interleaved with the timeslices of at least one other series, and an interrupt service routine including a thread scheduler invoked in response to each activation of said interrupt operation to determine whether a currently non-executing thread of instructions should preempt another thread of instructions then currently having control of the central processor, said multithreaded program comprising

a first thread of instructions executable by said central processor in successive increments during each of a first series of said timeslices, and

at least a second thread of instructions to take control of the central processor in response to determinations by said thread scheduler at each of those activations of said interrupt operation when said scheduler has determined that said second thread should preempt said first thread, so that said second thread is executed by and controls said central processor in successive increments during those second series timeslices selected by the scheduler in accordance with the respective determination at each interrupt operation,

whereby said program controls the single central processor to execute at least two threads of instructions concurrently with the successive execution increments of a thread interleaved with the successive execution increments of at least one other thread.

17. A computer memory element as set forth in claim 16 for use with a computer having an interactive user operable device and a video display monitor, and wherein

said second thread is responsive to said thread scheduler to preempt said first thread in response to each respective activation of said interrupt operation occurring immediately after each instance when the user has operated said user operable device,

said second thread comprises instructions executable interactively with a user in response to operation of said user operable device for interactively controlling the execution of said second thread,

said second thread further comprising instructions for activating said monitor to display the effects of said interactive operation of said user operable device,

whereby said second thread executes interactively with the user in the foreground while said first thread of the same program executes in the background concurrently with said second thread with control of the single central processor repeatedly switching back and forth between the threads so rapidly that an interactive user perceives the foreground and backgrounds to be executing simultaneously.

19. A storage means as set forth in claim 18 for use with a computer having a user operable device and a video display monitor, and wherein

one of said threads comprises instructions executable interactively with the user in the foreground in response to said user operable device for interactively controlling the execution of said one thread,

said one thread further comprising instructions for activating said monitor to display the effects of said interactive control by said user operable device,

another of said threads executing non-interactively in the background,

whereby said one thread executes interactively with the user in the foreground while said another thread of the same program executes in the background concurrently with said one thread with control of the microprocessor repeatedly switching back and forth between the threads so rapidly that an interactive user perceives the foreground and backgrounds to be executing simultaneously.

24. A disk store as set forth in claim 22 for use with a computer having a user-operable device for transmitting information from a user and a monitor screen for displaying information to the user, wherein

said another thread comprises executable instructions including means responsive to said user-operable device for entering said transmitted information and for interactively displaying on said screen the effects of said entered information, and

said one thread comprises executable instructions including means to process said entered information concurrently in real time as the information is being transmitted by said user-operable device,

whereby said another thread executes in the foreground interactively with and visibly to the user while concurrently therewith said one thread executes in the background so that the user perceives that both threads appear to be executing simultaneously whereas in reality the control of the central processor is alternately switching back and forth among the plurality of threads.

26. A computer-readable storage means encoded with executable instructions constituting computer operating software for preemptive multithreaded execution of a program having a plurality of instruction threads by a desktop computer having a microprocessor with an interrupt operation and a clock for periodically activating said interrupt operation, said software comprising:

thread scheduling means responsive to each activation of said interrupt operation to take control of the microprocessor from one of said threads and thereby preempt the execution of said one thread and to pass control of the microprocessor to another of said threads of said program for execution of said another thread, and

means operable after each execution of said another thread to return control of the microprocessor from said another thread to said one thread for resumption of the execution of said one thread at the point where it was preempted,

whereby at least two threads of the same program are concurrently active with said thread scheduling means periodically switching control of the same microprocessor preemptively from said one thread to said another thread so that the user perceives that both threads of instructions appear to be executing simultaneously.

27. A computer-readable storage disk encoded with executable instructions constituting operating software for preemptive multithreaded execution of a program having a plurality of concurrently executable instruction threads for processing the same body of data by a computer including a microprocessor having an interrupt operation, memory means for storing a body of data, input means for initiating execution of one of said instruction threads to process said stored body of data, and clock means for periodically activating said interrupt operation, said operating software comprising:

first software means responsive to periodic activation of said interrupt operation to cause the microprocessor to preemptively interrupt execution of said one thread and to execute another of said threads of said program whereby said another thread processes said body of data while the body of data is stored in said memory means, and

second software means to cause the microprocessor to discontinue execution of said another thread and to resume execution of said one thread so as to resume processing said stored body of data at the point in the body of data where said one thread was previously interrupted,

whereby said threads of the same program execute concurrently on the same microprocessor with said clock means periodically activating said interrupt operation to cause said microprocessor to preemptively switch execution from said one thread to said another thread periodically so that the user perceives that both threads appear to be executing simultaneously to process the same body of data.

28. A disk as set forth in claim 27 for use with a computer having a user-operable device for transmitting information from a user and a monitor screen for displaying information to the user, wherein

said another thread comprises instructions executable in response to said user-operable device for entering said transmitted information and for interactively displaying on said screen the effects of said entered information, and

said one thread comprises instructions executable to process said entered information concurrently in real time as the information is being transmitted by said user-operable device,

whereby said another thread executes in the foreground interactively with and visibly to the user while concurrently therewith said one thread executes in the background so that the user perceives that both threads appear to be executing simultaneously notwithstanding that control of the central processor is alternately switching back and forth among the plurality of threads of the same program.

30. A computer-readable disk memory having a surface formed with a plurality of binary patterns constituting a multithreaded word processing program executable by a computer having a keyboard, a buffer for storing words of a language, a central processor and a clock-driven periodically-activated interrupt operation providing a plurality of series of spaced timeslices with the timeslices of each series interleaved with the timeslices of at least one other series, said multithreaded word processing program comprising

a plurality of sets of instructions with each set executable to provide a respective thread of execution and with each thread having a respective task to perform,

a first of said sets of instructions executable to provide a first thread of execution having control of the central processor during each successive timeslice of a first series of timeslices with successive portions of the task of said first thread performed during respective successive timeslices of said first series,

said first set of instructions comprising lexical means for checking the spelling of words stored in said buffer,

said first thread of execution being periodically preempted at each termination of a timeslice of said first series by said clock activation of said interrupt operation, and

a second of said sets of instructions executable to provide a second thread of execution and responsive to said periodic preemptions to acquire control of the central processor during successive timeslices of a second series of timeslices with successive portions of the task of said second thread performed during respective succesive timeslices of said second series,

said second set of instructions comprising editor means responsive to said keyboard

for entering words into said buffer,

whereby a preemptive multithreading mode of operation is provided for the concurrent execution of said instruction threads with said lexical means checking the spelling of said entered words in real time while concurrently therewith said editor means is entering additional words into said buffer.

31. A disk memory as set forth in claim 30 for use in a computer also having a video display, and wherein

said editor means is responsive to a sequence of keyboard keystrokes at spaced time intervals,

said lexical means executing during time intervals between keystrokes,

each keystroke of a subset of said keykeystrokes corresponding to a respective alphanumeric character, and

means responsive to each of said alphanumeric keystrokes for immediately displaying on said video display the respective alphanumeric character corresponding to said keystroke,

whereby said first thread of execution checks the spelling of said words stored in said buffer in real time between keystrokes concurrently with the entry and display of said additional words by said second thread of execution.

33. A computer-readable element as set forth in claim 32 for use with a computer also having a video display, and wherein

said second thread includes instructions responsive to a sequence of keyboard keystrokes at spaced time intervals,

said first thread executing during time intervals between keystrokes,

each keystroke of a subset of said keykeystrokes corresponding to a respective alphanumeric character, and

said second thread includes instructions responsive to each of said alphanumeric keystrokes for immediately displaying on said video display the respective alphanumeric character corresponding to said keystroke,

whereby said first thread of execution checks the spelling of said words stored in said memory means in real time between keystrokes concurrently with the entry and display of said additional words by said second thread of execution.

34. A computer-readable disk memory having a surface formed with a plurality of binary patterns constituting a multithreaded application program executable by a computer having a keyboard, a buffer for storing data words, a central processor and a clock-driven periodically-activated interrupt operation providing a plurality of series of spaced timeslices with the timeslices of each series interleaved with the timeslices of at least one other series, said multithreaded application program comprising

a plurality of sets of instructions with each set executable to provide a respective thread of execution and with each thread having a respective task to perform,

a first of said sets of instructions executable to provide a first thread of execution having control of the central processor during each successive timeslice of a first series of timeslices with successive portions of the task of said first

thread performed during respective successive timeslices of said first series,

said first set of instructions comprising means for processing said data words stored in said buffer,

said first thread of execution being periodically preempted at each termination of a timeslice of said first series by said clock activation of said interrupt operation, and

a second of said sets of instructions executable to provide a second thread of execution and responsive to said periodic preemptions to acquire control of the central processor during successive timeslices of a second series of timeslices with successive portions of the task of said second thread performed during respective successive timeslices of said second series,

said second set of instructions comprising means responsive to said keyboard for entering data words into said buffer,

whereby a preemptive multithreading mode of operation is provided for the concurrent execution of said instruction threads with said first thread processing said entered data words in real time while concurrently therewith said second thread is entering additional data words into said buffer.

40. A computer-readable storage element containing an interactively multithreading application program having a plurality of threads of instructions executable to control the operation of a desktop computer including a microprocessor, a memory, a keyboard, an interrupt, a clock periodically activating the interrupt, an interrupt service routine responsive to each activation of the interrupt to preempt a currently executing thread and to take control of the microprocessor and to determine whether the keyboard has been struck, and a video monitor,

said application program comprising:

a first set of instructions executable by said microprocessor to provide a first thread of execution,

at least a second set of instructions executable by said microprocessor to provide at least a second thread of execution,

the execution of said first thread being periodically preempted at predetermined equally spaced time instants by said interrupt service routine in response to repeated activations of said interrupt by said clock,

said second thread being responsive to invocations by said interrupt service routine to take control of the microprocessor at each of those activations of said interrupt operation when said interrupt service routine has determined that the keyboard has been struck,

said second thread including means for entering a code into the memory in response to each keystroke of a sequence of successive keystrokes on said keyboard with a time interval between the keystrokes of each pair of successive keystrokes,

said second thread also including means for displaying the entered code on said video monitor immediately in response to each keystroke on said keyboard,

said first thread including means for processing said entered code,

whereby said threads execute concurrently in an interleaved relation with said first thread repeatedly processing said entered code during said time intervals between successive keystrokes.

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L12: Entry 3 of 6 File: USPT Oct 16, 2001

DOCUMENT-IDENTIFIER: US 6304891 B1
TITLE: Execution control for processor tasks

Brief Summary Text (6):
One approach to task management for processes which need to be completed in a specified interval of time is to divide time into a discrete series of units known as "frames." Frames are intervals of time in which an interrupt or other timing signal is generated to a processor at regular intervals and each of the tasks being executed by the processor is serviced in sequence. In such a frame-based processing system, each of the tasks is typically linked or associated with one another through some data structure, and the data structure is traversed during the servicing of the interrupt at the beginning of the frame, such that each task is serviced within the frame. A frame length is typically chosen based upon available cache memory in the system, and the minimum possible rate at which specific tasks should be serviced, among other considerations. For instance, a MIDI application (one using the Musical Instrument Digital Interface) requires minimum frame duration of 2 to 4 milliseconds. Applications using the V.32 data modem requires a maximum frame limit of 13 milliseconds. At any rate, frame size is typically driven by the application, available hardware, and other factors.

Current US Original Classification (1):
718/107

Previous Doc Next Doc Go to Doc#

L24: Entry 1 of 72 File: PGPB Nov 22, 2001

DOCUMENT-IDENTIFIER: US 20010043572 A1
TITLE: METHOD AND APPARATUS FOR MULTIPLE ACCESS COMMUNICATION

Application Filing Date:
19980924

Summary of Invention Paragraph:
[0004] A variety of techniques are known for allowing multiple users to communicate with one or more fixed stations (i.e., base stations) by making use of shared communication resources. Examples of multiple access communication systems include, for example, cellular telephone networks and local wireless communication systems, such as wireless private branch exchange (PBX) networks. In such multiple access communication systems, transmissions from different sources may be distinguished in a variety of manners, such as on the basis of different frequencies, time slots, and/or codes, for example.

Summary of Invention Paragraph:
[0005] As used herein, a communication system in which transmissions are distinguished according to the transmission frequency may be referred to as a frequency division multiple access (FDMA) communication system. A communication system in which a forward link transmission over one frequency is paired with a reverse link transmission over a different frequency may be referred to as a frequency division duplex (FDD) communication system.

Summary of Invention Paragraph:
[0014] In a second embodiment, a base station also comprises two base station sub-units. In the second embodiment, a time frame comprises a plurality of base transmit time slots defined with respect to a base transmit frequency band and a plurality of user transmit time slots defined with respect to a user transmit frequency band. The time frame is divided between the two base station sub-units such that the first base station sub-unit and second base station sub-unit each are assigned one half of the base transmit time slots and one half of the user transmit time slots. The base transmit time slots assigned to each base station sub-unit may form a contiguous block, or may alternate with one or more base transmit time slots assigned to the other base station sub-unit. Duplex communication channels are preferably defined by correlating a base transmit time slot with a user transmit time slot, with the base transmit time slots and user transmit time slot preferably separated by a sufficient amount of time to allow transmit/receive switching by a user station between the base transmit time slot and the user transmit time slot. Multiple time slots may be aggregated to a single user station in certain embodiments.

Detail Description Paragraph:
[0034] FIG. 2 is a diagram of a particular TDD frame structure as known in the art. In FIG. 2, a repeating major time frame 201 comprises a plurality of time slots (or minor time frames) 202. Each time slot 202 can be assigned by the base station 104 to a user station 102. User stations 102 can be assigned more than one time slot 202 if desired, and the time slots 202 so assigned may or may not be contiguous.

Detail Description Paragraph:

[0039] To facilitate rapid or convenient storage and extraction of data, the memory buffer 411 may be partitioned into memory segments 429, each memory segment 429 corresponding to one time slot 202. In one embodiment, for example, the current time slot (as output from, for example, the slot counter 422) can be used as a pointer offset to control which memory segment 429 the radio transceiver 405 is accessing at a given time. The memory segments 429 can be organized such that the data for the user transmission time segment 206 and data for the base transmission time segment 205 are stored adjacent to one another. Alternatively, the memory segments 429 can be organized such that the data for all of the user transmission time segments 206 are stored in one half of the memory buffer 411, and the data for all of the base transmission time segments 205 are stored in the other half of the memory buffer 411. In such a case, the control signal for the T/R switch 417 can be used as an additional pointer offset to control whether the radio transceiver 405 will access the "upper" half of the memory buffer 411 or the "lower" half of the memory buffer 411 (i.e., the user transmission data or the base transmission data).

Detail Description Paragraph:
[0040] The base station 401 shown in FIG. 4 may also provide for selection of transmission and reception frequency, so as to allow deployment of the base station 401 in a cellular environment in which different cells 103 (see FIG. 1) are assigned a different frequencies (consistent with a repeating pattern, such as a three-cell or seven-cell repeating pattern, as disclosed, for example, in U.S. Pat. No. 5,402,413, incorporated herein by reference as if set forth fully herein). The base station 401 can be deployed with the desired frequency by, for example, selecting external switches on the base station 401 or programming the desired frequency using software or firmware of the over-the-air controller 410. In the base station 401 shown in FIG. 4, the radio transceiver 405 comprises a programmable voltage-controlled oscillator 418, which is responsive to a control signal (e.g., control bits) from the over-the-air controller 410 and generates an output frequency according to such a control signal. Because the base station 401 implements a TDD time frame 201 such as shown in FIG. 2, it uses the same frequency for transmission and reception.

CLAIMS:

13. The method of claim 11, wherein said first half of said time frame and said second half of said time frame each comprise non-contiguous portions of said time frame, such that base transmit time slots of said first plurality of duplex time slots are interleaved with base transmit time slots of said second plurality of duplex time slots, and user transmit time slots of said first plurality of duplex time slots are interleaved with user transmit time slots of said second plurality of duplex time slots.

23. A base station, comprising: a radio transceiver, said radio transceiver comprising a base transmitter and a base receiver selectably connected to at least one base antenna by a transmit/receive switch, and said radio transceiver further comprising a voltage controlled oscillator whereby a transmission/reception frequency of said radio transceiver is set; a memory buffer connected to said radio transceiver, said memory buffer partitioned into a plurality of memory segments according to separate time division duplex communication channels, one memory segment for each such time division duplex communication channel; an over-the-air controller connected to said radio transceiver, said over-the-air controller comprising a time frame counter and a time slot counter; a toggle signal output from said over-the-air controller to said transmit/receive switch, said toggle signal causing said base transmitter and base receiver to be alternately connected to said at least one base antenna; a frequency selection signal output from said over-the-air controller to the voltage controlled oscillator of said radio transceiver, said frequency selection signal causing said radio transceiver to alternate between a base transmit frequency band and a base receive frequency band,

said base transmit frequency band being selected when said base transmitter is connected to said at least one base antenna, and said base receive frequency band being selected when said base receiver is connected to said at least one base antenna; and a backhaul interface connected to said memory buffer, said backhaul interface multiplexing information from said memory buffer for transmission over a backhaul line, and demultiplexing information received over said backhaul line for storage in said memory buffer.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L12: Entry 4 of 6 File: USPT Nov 21, 2000

DOCUMENT-IDENTIFIER: US 6151538 A
TITLE: Control system

Brief Summary Text (4):
Hybrid control systems are control systems involving both computer hardware and software as wall as other types of hardware such as sensors. An aircraft engine controller, for example, takes inputs such as engine speed, temperature and pressure; processes this information; and them provides outputs for controlling the operation of the engine, such as a signal indicating that more or less fuel is required. It is necessary for such a control system to execute various steps at set frequencies and in a particular order. For example, the above control system might input a pressure value, carry out a first processing step on that pressure value and carry out a second processing step on the results of the first processing step, before finally outputting a command to the engine depending on the results of the processing steps. It might then be necessary to repeat this whole sequence at a regular fixed interval. The steps in this sequence must execute at the correct frequencies and in the correct order to ensure that the engine is controlled in the desired manner.

Current US Cross Reference Classification (4):
718/107

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Search Forms
Search Results
Help
User Searches
Preferences
Logout

Generate Collection

L20: Entry 4 of 5 File: USPT Jun 9, 1992

DOCUMENT-IDENTIFIER: US 5121390 A
TITLE: Integrated data link controller with synchronous link interface and asynchronous host processor interface

Application Filing Date (1):
19900315

Brief Summary Text (25):
The asynchronous section comprises a DMAC (Direct Memory Access control) partition, a Master I/O Partition (MIO), and a Slave I/O Partition (SIO). The DMAC operates in association with the MIO to access memory in external higher level processing systems, in a direct access mode, for exchanging communication data and device control information between memories in the device and the external memories. Through featured actions discussed below, the external system places programmable control parameters for this partition in an associated DMA RAM memory (DMAR) in the device and in a FIFO RAM (FIFOR).

Brief Summary Text (30):
Communication data passing between the synchronous and asynchronous sections is held in a RAM memory, called FIFOR (First In First Out RAM), which is accessible to both sections. Data received from the network and processed through the synchronous section is placed in FIFOR a byte at a time, and data being processed for transmission to the network is fetched from FIFOR a byte at a time. Data being processed by the DMAC partition in the asynchronous section, for external transfer to or from external memory, is read from or written to FIFOR a word at a time.

Brief Summary Text (36):
The asynchronous section contains separate data communication and control interfaces to the external system bus. The data communication interface is used by the above-mentioned DMAC partition for transferring communication data between external system memory and FIFOR, and also for transferring command blocks for controlling the DMAC between the external memory and both DMAR and FIFOR. The control interface, which operates through the SIO (Slave I/O) partition mentioned earlier, is controllable by external systems, and usable by the latter for transferring control information to the device, and for retrieving interruption requests and device status information from the device.

Drawing Description Text (4):
FIG. 3 is a high level block diagram of the internal logical organization of the subject device illustrating its major logical partitions and associated RAM memory units including TS RAM.

Detailed Description Text (57):
FIG. 3, a block diagram of the logical organization of subject IDLC device, illustrates positioning of the device between IOP bus 35 and IF 27 previously discussed, and division of the device circuitry into synchronous and asynchronous sections, 50 and 51 respectively. This sectioning is of particular interest presently. Also shown are principal logical partitions in each section, each partition understood to contain one or more autonomous elements to be shown and

described later. Section 50 operates in synchronism with appearances of channel times slots at IF 27 to perform operations relative to respective communication channels. Operations in section 51 relative to any channel are performed in asynchronous relation to respective time slot appearances at IF 27. RAM (random access memory) units 52 and 53 are accessible to partitions in both sections, whereas RAM unit 54 is accessible only to partitions in the asynchronous section. The synchronous section contains logical partitions 60-64, and the asynchronous section contains partitions 65-67.

Detailed Description Text (65):
FIFO management partition 62 interfaces between FIFO RAM 52 (later abbreviated as FIFOR) and the receive and transmit partitions to transfer receive and transmit data between per channel queues in the RAM and these partitions on a FIFO (first in first out) basis, and to manage the utilization of such queues. Partition 62 monitors the state of occupancy of each channel queue in RAM 52, and supplies requests to DMA (Direct Memory Access) partition 65 in the asynchronous section which evoke actions through master I/O (MIO) partition 66 causing data to be transferred between respective queues and host system memories via bus 35. The requests from partition 62 to partition 65 are actually latched by a separate DMA Request Queue (DMARQ) partition not shown in FIG. 3, and handled by the latter on an asynchronous basis.

Detailed Description Text (293):
The BTDM contains data and control lines having relative signal timings illustrated in FIGS. 13 and 14. Single lines, RDATA and TDATA, respectively carry received and transmitted data (received data from L1 to IDLC, transmitted data from IDLC to L1). Control lines LSOF and BOS transfer reference timing pulses, from L1 to IDLC, respectively indicating "last slot of frame" and "beginning of slot" points of time relative to TDM time frames and time slots. LSOF is both the beginning of slot indication for the last frame slot and an advanced end of frame indication. Durations of pulses on LSOF and BOS are 50 nanoseconds or one (IDLC, L1) machine clock cycle. Each frame slot is allocatable to a communication channel, and each communication channel may be dynamically allocated one or more slots per frame. Active B, D, and clear/voice channels, are each allotted one slot per frame, and several slots (not necessarily contiguous in time) may be allotted collectively to a single HyperChannel (see description below of HyperChannels).

Detailed Description Text (416):
FIFOR also provides buffer storage of control word information used by the FIFO managers. Such information is written to FIFOR by SIO and DMAC. Like TSR, FIFOR contains a RAM memory array and logic for directing flow of information between that array and other partitions.

Detailed Description Text (433):
When an RDCR2 space becomes filled with 4 bytes, RFM posts a request to the DMARQ (DMA Request Queue) which causes the DMAC partition operating (asynchronously) in association with MIO partition to transfer the 4 bytes to a space in external (IOP/host) memory designated by address information in the respective RDCR1 field named RDCA (Receive DMA Current Address; 22 bits). The external memory is addressed as a circular buffer (refer to DMAC detailed description which follows). The remaining 10 bit spaces in RDCR1 consist of 8 reserved bit spaces (RES), a 1-bit space RPE (Receive Parity Error) for indicating parity error detected at the FIFOR output to DMAC, and a 1-bit space RBC (Receive Boundary Check) for indicating boundary check conditions encountered during the transfer (refer to DMAC description).

Detailed Description Text (1013):
If a valid address for a partition was decoded, enable select line (SIO.sub.-- DMACR.sub.-- RD, SIO.sub.-- DMACR.sub.-- WR, SIO.sub.-- FIFO.sub.-- RD, SIO.sub.-- FIFO.sub.-- WR, SIO.sub.-- TSR.sub.-- RD, SIO.sub.-- TSR.sub.-- WR, SIO.sub.--

RSM.sub.-- RD, or SIO.sub.-- RSM.sub.-- WR) to appropriate partition, based on the address decoded and the state of IOP.sub.-- RD/WR signal; go to state 3.

Detailed Description Text (1287):
The device also allows for synchronous storage of event status information associated with said requests in a memory shared for storing control parameters governing the processing activities in said pipeline. The device also allows for queued maintenance of event status in said memory, relative to plural time spaced events occurring within a communication channel in said shared memory, under the management direction of said interrupt handling partition. Finally, the device includes a slave I/O partition allowing the IOP to asynchronously access and retrieve said queued status with minimal interference to ongoing data communication processes.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L20: Entry 2 of 5 File: USPT Jun 8, 1993

DOCUMENT-IDENTIFIER: US 5218680 A
TITLE: Data link controller with autonomous in tandem pipeline circuit elements relative to network channels for transferring multitasking data in cyclically recurrent time slots

Application Filing Date (1):
19900315

Brief Summary Text (16):
The subject device is organized into synchronous and asynchronous sections (refer to the first application listed above under "Cross References To Related Patent Applications"), each containing multiple partitions of special purpose logic circuits and memory facilities. The foregoing objectives are achieved by organizing the partitions into modular autonomous units, hereafter called "autonomous logic elements", forming distributed logical pipelines relative to the handling of communication data between network and host interfaces. These distributed pipeline configurations provide advantages relative to contemporary systems, in respect to numbers of logical functions which can be performed in parallel simultaneously relative to any network channel, and also in respect to design adaptiveness (any element can be replaced by a functionally different element, e.g. for supporting future links requiring new functions, without affecting designs of other elements in the same pipeline), and the pipeline organization adapts efficiently to interpolation of additional elements in tandem into any pipeline.

Brief Summary Text (20):
Another aspect of the device is that it contains interrupt handling partitions and a DMA (direct memory access) control partition, the latter having direct access to memory in the host processing system, characterized in their autonomy relative to host system elements and their offloading of operations ordinarily performed at a higher processing level.

Brief Summary Text (21):
Another feature is that the DMA control partition acts through a Master I/O partition in accessing the bus through which access to host system memory is obtained. The Master I/O partition performs the arbitration functions required to access the bus, and provides a small body of logic as a focal point for adapting the subject device design for interfacing to a variety of different bus structures.

Brief Summary Text (23):
One of the interrupt handling partitions (INT) monitors events within the device and at the interface between the device and the network (including hardware conditions and activities relative to link channels), collects relevant status details in local memory within the device and sets alerting indications in the other partition (SIO). Processing elements in the host system monitor the SIO for such indications and operate through the SIO to directly access local memory in the device to collect the stored status details. This eases the time criticality of reportage of time related events.

Brief Summary Text (45):
Another object is to provide a communication controller, using a device as characterized above, having an autonomous interrupt handling partition and local memory operative to collect status information for delayed communication to the host system.

Brief Summary Text (46):
Another object is to provide a communication controller, using a device as characterized above, having an autonomous DMA control partition for directly accessing memory in an external host processing system, and an associated local RAM for storing DCB control block information supplied by said system for use in transmitting data from said host system to said network channels, wherein said partition is responsive to chaining indications in said DCB's to retrieve additional DCB's from scattered spaces in said host system memory and said device is operative further to determine network framing and formatting of transmitted data for varied protocol channels in accordance with control information in said additional DCB's.

Drawing Description Text (4):
FIG. 3 is a high level block diagram of the internal logical organization of the subject device illustrating its major logical partitions and associated RAM memory units including TS RAM.

Detailed Description Text (58):
FIG. 3, a block diagram of the logical organization of subject IDLC device, illustrates positioning of the device between IOP bus 35 and IF 27 previously discussed, and division of the device circuitry into synchronous and asynchronous sections, 50 and 51 respectively. This sectioning is of particular interest presently. Also shown are principal logical partitions in each section, each partition understood to contain one or more autonomous elements to be shown and described later. Section 50 operates in synchronism with appearances of channel times slots at IF 27 to perform operations relative to respective communication channels. Operations in section 51 relative to any channel are performed in asynchronous relation to respective time slot appearances at IF 27. RAM (random access memory) units 52 and 53 are accessible to partitions in both sections, whereas RAM unit 54 is accessible only to partitions in the asynchronous section. The synchronous section contains logical partitions 60-64, and the asynchronous section contains partitions 65-67.

Detailed Description Text (66):
FIFO management partition 62 interfaces between FIFO RAM 52 (later abbreviated as FIFOR) and the receive and transmit partitions to transfer receive and transmit data between per channel queues in the RAM and these partitions on a FIFO (first in first out) basis, and to manage the utilization of such queues. Partition 62 monitors the state of occupancy of each channel queue in RAM 52, and supplies requests to DMA (Direct Memory Access) partition 65 in the asynchronous section which evoke actions through master I/O (MIO) partition 66 causing data to be transferred between respective queues and host system memories via bus 35. The requests from partition 62 to partition 65 are actually latched by a separate DMA Request Queue (DMARQ) partition not shown in FIG. 3, and handled by the latter on an asynchronous basis.

Detailed Description Text (293):
The BTDM contains data and control lines having relative signal timings illustrated in FIGS. 13 and 14. Single lines, RDATA and TDATA, respectively carry received and transmitted data (received data from L1 to IDLC, transmitted data from IDLC to L1). Control lines LSOF and BOS transfer reference timing pulses, from L1 to IDLC, respectively indicating "last slot of frame" and "beginning of slot" points of time relative to TDM time frames and time slots. LSOF is both the beginning of slot indication for the last frame slot and an advanced end of frame indication.

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, September 27, 2004

Hide?	Set Name	Query	Hit Count
	DB=USPT; PLUR=YES; OP=ADJ		
☐	L31	709/107.ccls.	0
☐	L30	709/107.ccls.	0
☐	L29	709/107.ccls.	0
	DB=PGPB,USPT; PLUR=YES; OP=ADJ		
☐	L28	709/107.ccls.	0
☐	L27	709/103.ccls.	0
☐	L26	709/102.ccls.	0
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L25	709/102.ccls.	0
☐	L24	l22 NOT l23	72
☐	L23	L22 and l3	2
☐	L22	L21 NOT l20	74
☐	L21	L19 and ((more or different or various) near4 (rate or frequency))	76
☐	L20	L19 and ((valid or validity) near5 (software or partition or region))	5
☐	L19	L7 and l2	208
☐	L18	L17 and l2	2
☐	L17	L16 and l10	575
☐	L16	l7 and background and foreground	686
☐	L15	l7 and backgroung and foreground	0
☐	L14	l7 and l10 and backgroung and foreground	0
☐	L13	l4 and backgroung and foreground	0
☐	L12	L11 and backgroung and foreground	0
☐	L11	L10 and l8	126
☐	L10	(software or task)	650149
☐	L9	L8 and l3	3
☐	L8	L7 and l2	208
☐	L7	memory near8 (partition or region or partitioning or partitioned)	57625
☐	L6	l4 NOT l5	37
☐	L5	L4 and stack	5
☐	L4	L3 and l2	42

☐	L3	(repetitive or repetitively) near8 (execute or executing or executed)	3085
☐	L2	20000516	9274
☐	L1	(sequence or contiguous) near8 (time near4 (interval or slot or partition or partitioning or partitioned))	13922

END OF SEARCH HISTORY

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, September 27, 2004

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT; PLUR=YES; OP=OR		
☐	L12	L11 and l8	6
☐	L11	(sequence near8 (interval or slot)) same (rate or frequency)	4601
☐	L10	L9 and l8	1
☐	L9	sequence near8 (interval or slot) near8 (rate or frequency)	1638
☐	L8	(718/107.ccls. or 718/103.ccls. or 718/102.ccls. or 709/215.ccls. or 709/214.ccls.)	2081
☐	L7	(718/107.ccls. or 718/103.ccls. or 718/102.ccls. or 709/215.ccls. or 709/214.ccls.)	2081
☐	L6	(718/107.ccls.)	538
☐	L5	(718/103.ccls.)	425
☐	L4	(718/102.ccls.)	924
☐	L3	(709/215.ccls. or 709/214.ccls.)	425
☐	L2	(709/215.ccls.)	197
☐	L1	(709/214.ccls.)	266

END OF SEARCH HISTORY